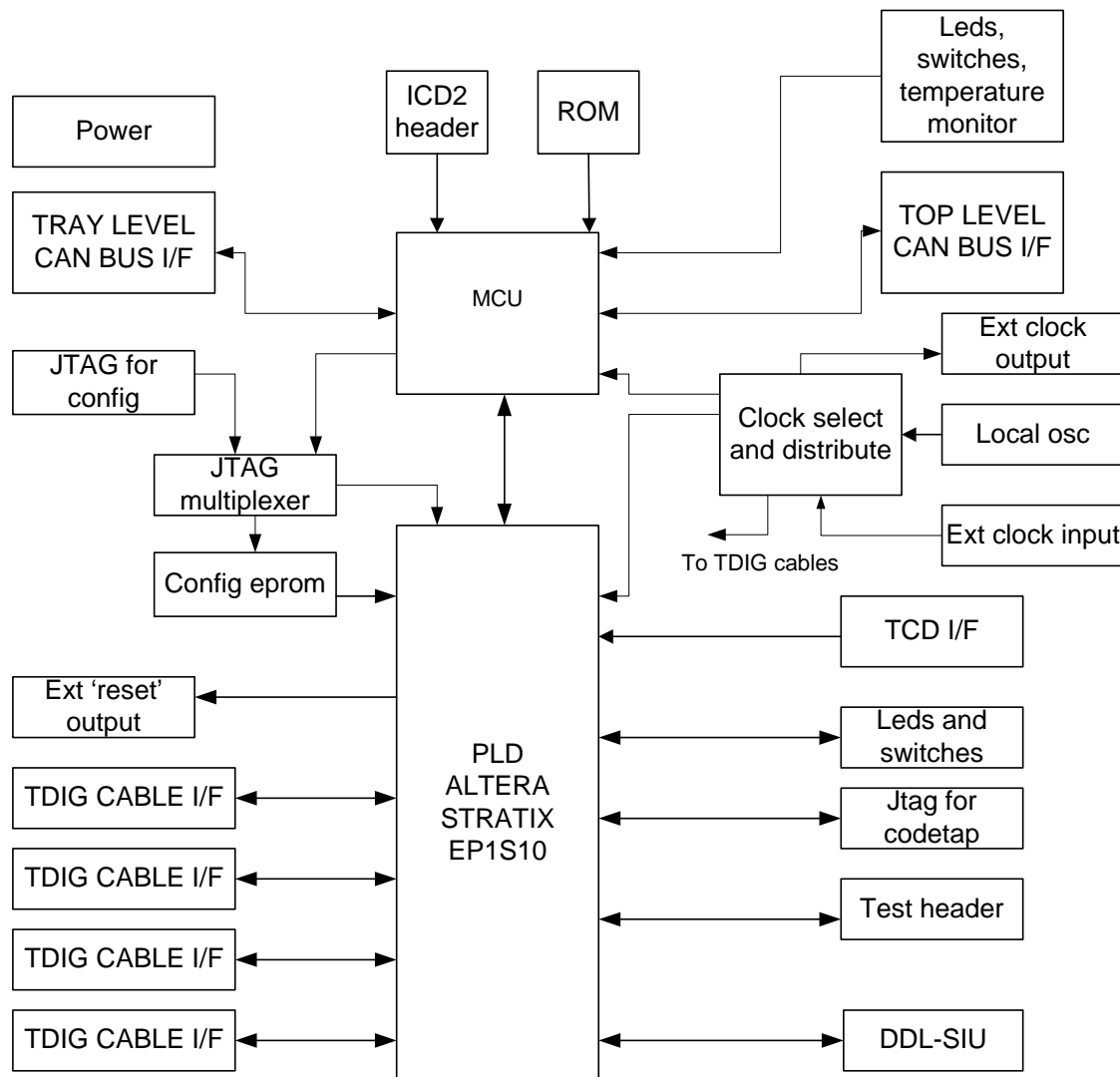


Overview

The TCPU generates system clocks and aggregates data from either start or stop detectors. It communicates with STAR DAQ via a DDL-SIU card, and it communicates with a local host PC over CAN bus.

Functional Modules

A top level block diagram for TCPU is shown below.



TCPU pld/mcu interface**1. Interface signals**

Signal name	MCU pin / direction	PLD pin / direction	PLD port	PLD signal
Data	RF(7..0) / inout	G4,G3,G2,G1,F22,F21,F18,F17 / inout	mcud (7 downto 0)	mcu_data (7 downto 0)
Adr	RG(4,3,0) / out	F14,F10,F9 / in	mcuctl (2 downto 0)	mcu_adr (2 downto 0)
!Read / Write	RC0 / out	F15 / in	mcuctl(3)	readbar_write
Fifo empty	RA1 / in	P9 / out	fifo_empty	fifo_empty
Data strobe	RB0 / out	AB19/ in	pld_int	data_strobe

Table 1 : TCPU interface signals between MCU and PLD

The readbar_write signal and the 3 bit address signals are decoded to give clock enable for registers that the mcu is writing to, or read enable for buffers/registers that the mcu is reading from.

MCU writes to PLD:

Address	Register
0	Mode (see mode definitions below)
1	Configuration (see configuration bit definitions below)
2	MCU filter (Select data for MCU fifo - see table)
6	TDC reset – global reset to TCPU PLD.
7	TDC bunch reset – sent to TDIG boards over ribbon cable.

Table 2: Registers in PLD that are written by MCU**MCU reads from PLD:**

Address	Register
0	Mode readback
1	Configuration readback
2	FIFO 3 (most significant byte)
3	FIFO 2
4	FIFO 1
5	FIFO 0
6	FIFO read request
7	Status

Table 3: Buffers/registers in PLD that are read by MCU

Register definitions:

Mode register

Mode register value	Mode name	Actions by pld
X"00"	idle	Readout determined by Configuration register below
X"01"	normal	read all 4 cables + trigger
X"02"		
X"03"		
X"04"		
X"05"		

Configuration register

bit #	Meaning	default
7		
6	Readout Mask bit 2	000 = TCD, 001 = Cable 1
5	Readout Mask bit 1	010 = Cable 2, 011 = Cable 3,
4	Readout Mask bit 0	100 = Cable 4
3	Readout Source bit 2	000 = TCD, 001 = Cable 1,
2	Readout Source bit 1	010 = Cable 2, 011 = Cable 3,
1	Readout Source bit 0	100 = Cable 4
0	Ext trigger	0 = ext trig; 1 = internal trig

MCU filter select

Register value	Meaning	
7		
6		
5	mcu_fifo(7 downto 0)	
4	mcu_fifo(15 downto 8)	
3	mcu_fifo(23 downto 16)	
2	mcu_fifo(31 downto 24)	
1	config_data	
0	mode_data	

Draft TCPU operational manual Version 2

October 4, 2005

TDIG I/F

Signal Dictionary:

Signal	Justin's name	#signals	Differential	Level	Fresno IN OUT
Cn_DCLKN	data_clk	1	Y	LVDS	IN
Cn_M24N	mul24	1	Y	LVDS	IN
Cn_TDCN[3..0]	tdc_data[3..0]	4	Y	LVDS	IN
Cn_M7N[5..0]	mul7[5..0]	6	Y	LVDS	IN
Cn_TDC_TRIGN	tdc_trig	1	Y	LVDS	OUT
Cn_M24_TRIG	mul_24_trig	1	Y	LVDS	OUT
Cn_M7_GATE	rhic_strb	1	Y	LVPECL	OUT
Cn_SYS_CLK	4x_rs	1	Y	LVPECL	OUT
Cn_RS232_RX1	rs232_tx	1		RS232	OUT
Cn_RS232_TX1		1		RS232	IN
Cn_RS232_RX2		1		RS232	OUT
Cn_RS232_TX2		1		RS232	IN
Cn_RST_OUT	tcpu_reset	1		LVTTTL	OUT
Cn_SPARE_OUT	data_valid_us_cable	1		LVTTTL	OUT

DCLK clocks M7, M24, and TDC data into Fresno.

M24 is serial input of uncompressed multiplicity data. Word length is 24 bits.

TDC[3..0] is TDC data. 1 complete word is 32 bits, so 8 transfers are required.

SYS_CLK is 40 Mhz clock for TDCs and PLDs

Clock Distribution

All sample clocks are sourced from a single 40 Mhz oscillator. This clock is distributed from TCPU1 to TCPU2 and TCPU3. The TCPUs are configured via CAN messages to run from either external or internal clocks. TCPU1 will operate from an internal clock. TCPU2 and TCPU3 will operate from an external clock (received from TCPU1).

TCPUs will power up with internal clock enabled. After being switched to external clock, watchdog timers will detect the absence of an external clock, and in this case the TCPUs will automatically switch back to internal clock. Missing clock status will be available as status over CAN bus.

The TCPUs can also selectively use the on-board PLL with either the internal or external clock.

I need to verify with Ted that if we somehow end up using the RHIC clock as an external clock to all TCPUs, then we can multiply it by 4 in the PLL. If not, then we should design the next revision of the TCPU to allow this, just in case.

Trigger processing

The TCD board for the PicoTOF subsystem sends its outputs to a TCD fanout card. This fanout card sends trigger commands to TCPU1, TCPU2, and TCPU3.

The trigger commands are used to:

- generate a 'trigger' signal which goes to the TDCs and serves as a reference for the trigger matching window.
- control the high level DAQ interface – when to “build events” by buffering data in the output fifos, when to transmit this data, when to abort transmission of this data
- synchronize trigger-matched TDC data with other experimental data by tagging the TDC data “event groups” with the TCD trigger token value (by inserting the TCD trigger word into the data stream to DAQ.

Event building and DAQ link

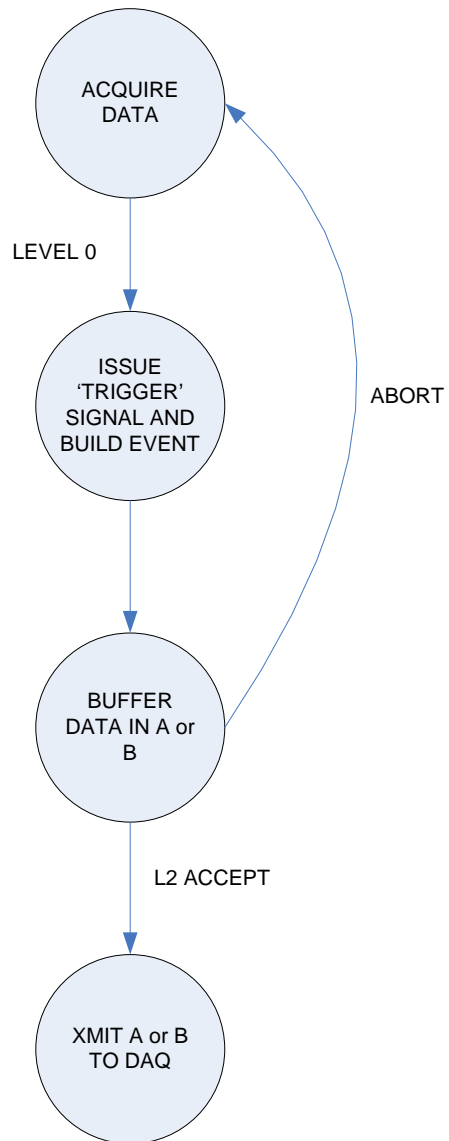
TDC words are combined into events, stored, and transmitted to DAQ with the following scheme:

- TDCs capture hits as they occur and store them in internal FIFOs
- When an L0 command is received from TCD, the TCPU issues 'trigger' signal to it's TDIG board(s).
- The TDCs perform trigger matching internally, forwarding only data within a time window relative to this 'trigger' signal.
- The TCPU receives TDIG data and builds an event and writes it to a readout buffer.
- Event size will be:

$$(\text{Occupancy} \times 2 \times 192 \text{ words}) + 5 \text{ header words}$$

- Readout buffer is a ping/pong buffer (A and B):
 - Start by writing to A.
 - If an L2 accept occurs, start writing A to DAQ. If an abort occurs, then clear A.
 - For subsequent events, write to A if A is empty, otherwise write to B. Clear or write the oldest data to DAQ (A or B).

Events are built and processed according to the following state diagram:



October 4, 2005

TCPU**Interfaces:**

- External trigger pulse **input**: differential LVPECL input at J39 (positive polarity) and J40 (negative polarity). This signal will be used as a “reset” from the master TCPU to reset all TDC bunch reset counters.
- Trigger pulse **output #1**: differential LVPECL input at J36 (positive polarity) and J35 (negative polarity).
- Trigger pulse **output #2**: differential LVPECL input at J38 (positive polarity) and J37 (negative polarity).
- Ribbon cables from 4 TDIG board pairs:

Signal	Justin's name	#signals	Differential	Level	Fresno IN OUT
--------	---------------	----------	--------------	-------	-----------------

Cn_DCLKN	data_clk	1	Y	LVDS	IN
Cn_M24N	mul24	1	Y	LVDS	IN
Cn_TDCN[3..0]	tdc_data[3..0]	4	Y	LVDS	IN
Cn_M7N[5..0]	mul7[5..0]	6	Y	LVDS	IN
Cn_TDC_TRIGN	tdc_trig	1	Y	LVDS	OUT
Cn_M24_TRIG	mul_24_trig	1	Y	LVDS	OUT
Cn_M7_GATE	rhic_strb	1	Y	LVPECL	OUT
Cn_SYS_CLK	4x_rs	1	Y	LVPECL	OUT
Cn_RS232_RX1	rs232_tx	1		RS232	OUT
Cn_RS232_TX1		1		RS232	IN
Cn_RS232_RX2		1		RS232	OUT
Cn_RS232_TX2		1		RS232	IN
Cn_RST_OUT	tcpu_reset	1		LVTTL	OUT
Cn_SPARE_OUT	data_valid_us_cable	1		LVTTL	OUT

DCLK clocks M7, M24, and TDC data into Fresno.

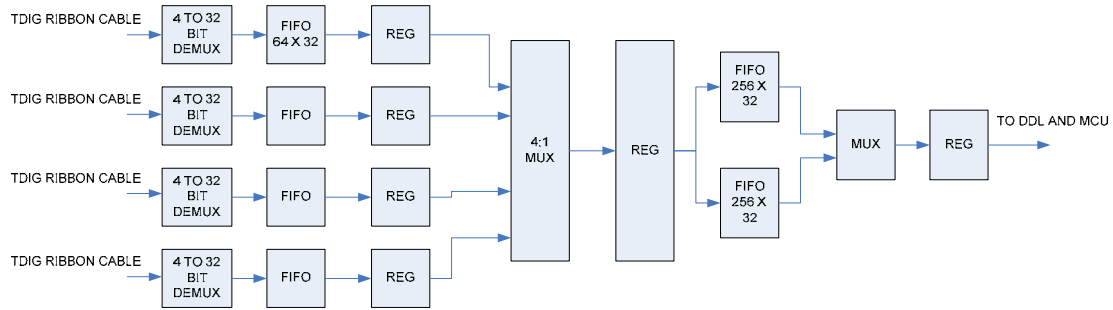
M24 is serial input of uncompressed multiplicity data. Word length is 24 bits.

TDC[3..0] is TDC data. 1 complete word is 32 bits, so 8 transfers are required.

SYS_CLK is 40 Mhz clock for TDCs and PLDs.

TCPU Data Flow

TCPU data flow is shown below. The 4-to-32 demux blocks receive 32 bit words over ribbon cable as 8 4bit packets. The reassembly is done automatically, and the data words are presented to the input FIFOs along with a synchronous strobe for the FIFO clock enable inputs. If the fifos are full, data will be lost.



All data paths are 32 bits. FIFOs have registered inputs and outputs. All registers have clock enable inputs. Data flow is controlled via mux select and clock enable signals.

A data flow controller circuit schedules data flow within this circuit. The controller:

- enables data clocking out of the input fifos
- selects which input stream will go to the output fifo
- selects which output fifo will receive input data
- selects which output fifo will transmit data

The data flow controller operates according to

- a mode register that is loaded from the MCU under canbus control
- the high level trigger command processing controller (see "event building and DAQ link" section above)

TCPU firmware architecture

The diagram below shows the module hierarchy in the TCPU PLD firmware.

